Stanford University Coursera Machine Learning Andrew Ng

Lecture 1. Introduction to Machine Learning

- Supervised learning vs unsupervised learning.
  ↓
  gives "right answer"
  for each example in data.

Lecture 2 Model Representation (linear regression with one variable) - Supervised learning.

- hypothesis $h_\theta(x) = \theta_0 + \theta_1 x$
- cost function. $J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} (\underbrace{h_\theta(x^{(i)})}_{\text{our prediction}} - y^{(i)})^2$

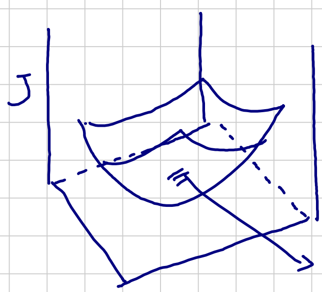
  training sample size

  ↳ goal: minimize J.

  actual data.

- θ_0, θ_1 are parameters → we choose the parameters to better fit our data to the answers.

- visualization



contour plots

min J)

<u>Gradient Descent</u>

- we have some function $J(\theta_0, \dots \theta_n)$, we want to $\min_{\theta_0 \cdots \theta_n} J(\theta_0, \theta_1)$.
- algorithm:

  repeat until convergence {

  $$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1) \quad ; \text{ for } \underbrace{j=0 \text{ and } j=1}_{\text{simultaneously update } \theta_0 \text{ and } \theta_1.}$$

  }

  learning rate.

e.g.

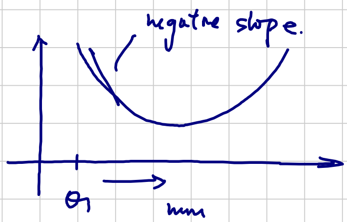
$$temp0 := \theta_0 - \alpha \frac{\partial}{\partial \theta_0} J(\theta_0, \theta_1)$$
$$temp1 := \theta_1 - \alpha \frac{\partial}{\partial \theta_1} J(\theta_0, \theta_1)$$

$$\theta_0 := temp0$$
$$\theta_1 := temp1$$

$\boxed{\text{intuition.}}$

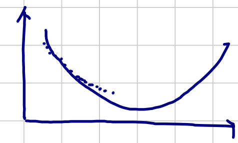
$$\theta_1 := \theta_1 - \underbrace{\overset{\alpha > 0.}{\alpha} \frac{\partial}{\partial \theta_1} J(\theta_1)}_{\text{if derivative} > 0}.$$

then θ_1 new $< \theta_1$, we are closer to the minimum point.

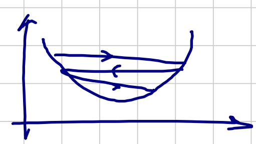


$\theta_1 := \theta_1 - \alpha \cdot (-ve \#)$, θ_1 increases.

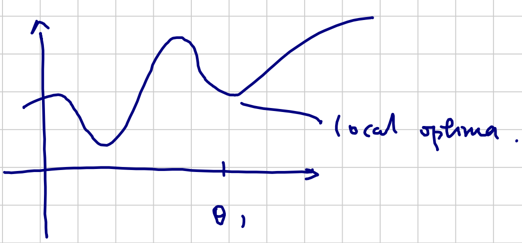
- if α is too small, gradient descent can be slow



- if α is too large, gradient descent can overshoot the minimum. It may fail to converge, or it could diverge.



Local minima



. gradient descent can converge to a local minimum, even when the learning rate α is fixed.

. we do not need to vary α.

observe: $\theta_1 := \theta_1 - \alpha \underbrace{\frac{\partial}{\partial \theta_1} J(\theta_1)}_{\text{scaling.}}$

as we approach a local minimum, gradient descent will automatically take smaller steps.

Gradient Descent for Linear Regression

Gradient Descent algorithm

repeat until convergence {

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1)$$

$\forall j = 0, 1$

}

Linear regression model.

$$h_\theta(x) = \theta_0 + \theta_1 x.$$

$$J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2$$

$$\Rightarrow \frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1) = \frac{\partial}{\partial \theta_j} \frac{1}{2m} \sum_{i=1}^{m} [\underbrace{h_\theta(x^{(i)})} - y^{(i)}]^2$$

$$= \frac{\partial}{\partial \theta_j} \frac{1}{2m} \sum_{i=1}^{m} (\theta_0 + \theta_1 x^{(i)} - y^{(i)})^2$$

$$j=0: \quad \frac{\partial}{\partial \theta_0} J(\theta_0, \theta_1) = \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}))$$

$$j=1: \quad \frac{\partial}{\partial \theta_0} J(\theta_0, \theta_1) = \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})) \cdot x^{(i)}$$

$\Rightarrow$ GDA (gradient descent algorithm)

repeat until convergence {

update θ_0 and θ_1 simultaneously.

$$\begin{cases} \theta_0 := \theta_0 - \alpha \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) \\ \theta_1 := \theta_1 - \alpha \frac{1}{m} \sum_{j=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) \cdot x^{(i)} \end{cases}$$

"Batch Gradient Descent"

↳ each step of gradient descent uses all the training samples. $\Rightarrow \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})$.